1 Introduction

The purpose of this work is to produce a part-of-speech tagger for French using morphological analysis provided by a finite-state transducer. The tagger also utilizes a combination of statistical learning and linguistic knowledge and is built in a pipelined architecture. All modules, except for preprocessing and morphological analysis, can be ordered in various ways. Part of speech tagging consists of applying several disambiguation modules on a list of ambiguous words until a single tag remains for each word. We propose and evaluate a sequencing strategy for the various modules and point out the best sequencing available. Several experiments were performed to figure out the best order of the different modules. Results showed that optimal results are obtained when morphological analysis is applied first, followed, in that order, by the application of linguistic constraints, the statistical stage, and, finally, the mapping of the large tagset to a smaller one.

2 Background

French is an inherently ambiguous language when it comes to morphological analysis. For example, the word "mise" can have as many as eight morphological analyses.

```
``mise'' - <mis>    adjective, feminine singular
``mise'' - <mis>    noun, feminine singular
``mise'' - <miser>  past participle, feminine singular
``mise'' - <miser>  verb, 1st person, singular, present, indicative
``mise'' - <miser>  verb, 1st person, singular, present, subjunctive
``mise'' - <miser>  verb, 2nd person, singular, present, imperative
``mise'' - <miser>  verb, 3rd person, singular, present, indicative
``mise'' - <miser>  verb, 3rd person, singular, present, subjunctive
```

The goal of a part-of-speech tagger is to reduce the number of part-of-speech ambiguities; this is achieved by using a combination of linguistic knowledge and statistical rules that progressively reduce the number of possible tags for a given word. A tag contains information about the part of speech, as well as about certain grammatical categories such as tense, mood, number, and gender. The input to the system is a French text, with 8-bit encoded accents. Table 1 shows an example of text data:

L'usine, qui devrait être implantée à Eloyes (Vosges) représente un investissement d'environ 3,7 milliards de yens (148 milliards de francs). Elle fabriquera, dans un premier temps, le produit liquide qui entre dans le processus des photocopies ainsi que des pièces détachées pour la filiale de Minolta en RFA.

Table 1: Corpus Sample of newswire compiled by the French embassy in Washington, D.C.

The goal is to obtain an output text where a single part-of-speech is associated with each word. Table 2 shows the output of the first sentence of the text in Table 1 disambiguated at a word level. In the left column are the words corresponding to the French corpus, the part-of-speech tags corresponding to the words ("tag" file) are in the right column.

Linguistic knowledge about possible sequencing of parts of speech is very powerful, since several types of restrictions can be expressed for words and tags in context. For example, an article cannot be followed by a verb in French, as well as in many other languages. Given that many words have unique tags, restriction rules could use such words as anchors to disambiguate surrounding words.

Word	p.o.s. tag	Meaning of the tag
$< S >$	^	beginning of sentence
L	RDF	definite feminine article
usine	NFS	feminine singular noun
,	.	punctuation
qui	E	relative pronoun
devrait	V3SPC	verb 3rd person singular present conditional
être	&N	auxiliary infinitive
implante'e	QSFS	past participle feminine singular
à	P	preposition
Éloyes	U	proper noun
(	.	punctuation
Vosges	U	proper noun
)	.	punctuation
représente	V3SPI	verb 3st person singular present indicative
un	RIMS	indefinite masculine singular article
investissement	NMS	masculine singular noun
d	P	preposition
environ	P	preposition
3	W	numeral
,	.	punctuation
7	W	numeral
milliards	W	numeral
de	P	preposition
yens	NMP	masculine plural noun
(	.	punctuation
148	W	numeral
milliards	W	numeral
de	P	preposition
francs	NMP	masculine plural noun
)	.	punctuation
$< S >$	$	end of sentence

Table 2: Sample output of the tagger for the first sentence from the text in Table 1

On the other hand, statistical learning is used as follows: given a manually tagged training corpus, the most frequent tags from each combination of tags can be easily learned. When the statistical knowledge is applied, the best decisions based on the disambiguated data are made.

We look at the morphological analysis, the deterministic stage, and the statistical stage as operators which modify the current tag assignment of the corpus and produce a new and more accurate tag assignment. There are additional modules, such as preprocessing and morphological stages, that are applied in a fixed order. The whole process of tagging can be looked at as the composition of these processing operators. Since the operators are compositional (they can be applied in any order), we can theoretically order them in many different ways. We want to find out what sequence of operators leads to an overall improvement of the tagging accuracy.

3 Related Work

There are a number of taggers and tagging methods available; for the last decades, works in part-of-speech tagging have generally followed either a rule-based approach ([9] , [4] , [15]), or a statistical one ([1] , [10] , [11] , [6] , [5]). Statistical approaches often use Hidden Markov Models for estimating lexical and contextual probabilities, while rule-based systems capture linguistic generalities to express contextual rules. Most of these works have benefited from large tagged corpora, making feasible the training and testing procedures. However, no publicly available large tagged corpora exist for French, so other techniques had to be discovered to tackle this problem.

4 Theoretical Principles

This section contains a formal description of the tagging scheme. A list of definitions of terms used in this work is also provided.

4.1 Definitions

- The **initial tag assignment** is the tag assignment after preprocessing and morphology.
- A **tag assignment** TA is a list of lexemes along with a set of tags (correct or not) assigned at a particular stage to each of the words in the corpus. The following is an example of a TA data structure:

$$\begin{pmatrix} L' & [BD3S & RDF & RDM] & & & \\ usine & [NFS & V1SPI & V1SPS & V2SPM & V3SPI & V3SPS] \end{pmatrix}$$

  The word to be tagged is in the left-hand column, whereas the the right-hand column lists the tags associated with this word. The left hand side is the word and the right hand side – the list of tags associated with this word.

- The **correct tag assignment** TA_c is a tag assignment in which each word has been assigned one tag only - the correct one. A training corpus of 10,000 words has been manually tagged and used as a basis for evaluating newly tagged corpora.
- The **tagsets** TS: two tagsets have been considered - a large one consisting of 253 tags, and a smaller one consisting of 67 tags. In addition, the user can specify any generalized subset of tags occuring in the large tagset. The tagsets are shown in Appendix A - Section C. The tags within each tagset have a hierarchical structure. They contain information about the

part of speech as well as some morphological features such as mood, tense, person, gender, and number. Each tag is actually an acronym carrying morphological information.

Example: V refers to verbs in general, $V3S$ refers to third person singular verbs of any mood or tense, $V3SPI$ refers to third person singular verbs in present of the indicative, and $V.S$ refers to all singular verbs.

This terminology has several advantages. When negative constraints are applied, they can be invoked at several levels of the tag, using all the available combinations; in the above example, a rule can apply to the part-of-speech (p.o.s.) only, the p.o.s. and the number, the p.o.s. and the person, the p.o.s. and the tense, or the p.o.s. with its mood, tense, person, and number.

- The **accuracy function** $c(TA_i)$ refers to the accuracy of the current tag assignment TA_i, when compared to the correct TA, i.e. (TA_c).
- The **inaccuracy function** $i(TA_i)$ refers to the percentage of incorrect tags in a given TA_i.
- The **ambiguity function** $a(TA_i)$ refers to the percentage of incorrect tags in a given TA_i.

  If TA_c has 1000 words, and 700 of them are tagged correctly in TA_i, 10 of them are tagged incorrectly, and the remaining 290 are still ambiguous at this stage, then $c(TA_i) = 70.0$ %, $i(TA_i) = 0.1$ %, and $a(TA_i) = 29.9$ %.
- A **genotype** is the list of all possible tags that a given word can inherit from the morphological module.

  Example: the word "*mise*" has the following genotype: [JFS NFS QSFS V1SPI V1SPS V2SPM V3SPI V3SPS].
- A **statistical decision** consists of a genotype, its most likely (predominant) resolution in the training corpus, and the likelihood of that resolution.

  Example: if $[PNP]$ are possible tags, then $[P]$ is selected in 96.85 % of the cases (768 out of 793).
- **Processing operators** are essentially functions that take a tag assignment as argument and produce another tag assignment. Operators are explained in more detail in the next section.

  Example: If P is a processing operator, and TA_1 a tag assignment, then $P(TA_1) = TA_2$, which means that TA_2 is the resulting tag assignment.

$$P\begin{pmatrix} L' & [BD3S & RDF & RDM] & & & \\ usine & [NFS & V1SPI & V1SPS & V2SPM & V3SPI & V3SPS] \end{pmatrix} = \begin{pmatrix} L' & [RDF] \\ usine & [NFS] \end{pmatrix}$$

- A **tagging scheme** is a composition of n processing operators, which, when applied on the initial tag assignment (TA_0) returns another tag assignment (TA_n). In order to keep our notation consistent, we shall use the concatenation of the symbols, representing the operators in composition to refer to a given tagging scheme. For example, we shall use **DAT** to express that 3 operators deterministic (D), application of n-gram statistical decisions(A), and tagset reduction (T) have been applied to the initial TA.

  For simplicity, the P, M, and L stages (preprocessing, morphology, and learning - see next section) will be omitted when referring to a particular tagging scheme. The rules are simple: P and M are applied first, also, there must be an L stage before the A stages.

  Example: The tag scheme DA_5DT means the composition $T(D(A_5(D(TA_0))))$.

- **Negative constraints**

  Negative constraints are examples of deterministic knowledge that express linguistic relationships between the features of the words in a given n-gram, thereby performing some contextual disambiguation over strings of tags. It seemed natural to use human expertise to partly disambiguate text through rules. Of course, one could argue that the machine would eventually learn it, but generalities that capture gender and number agreement are straightforward to state. They are available to the human without effort, they are easy to implement.

  Each of the linguistic constraints is applied several times over the anchors of the corpus. This way, anchors can create new anchors and thus enlarge the islands of disambiguated words.

  Example:

  - **BS3 BD1** (3rd person subject personal pronoun; 1st person indirect personal pronoun). In the phrase "il nous faut" ("we need", literally "it is necessary to us") – the tags are BS3MS for "il" and (BD1P BI1P BJ1P BR1P BS1P) for "nous". The negative constraint "BS3 BD1" rules out "BD1P" and thus reduces the alternatives from 5 to 4 for the word "nous".
  - **N K** (noun; interrogative pronoun). In the phrase "... fleuve qui ..." (...river, that...), "qui" can be tagged both as an "E" (relative pronoun) and a "K" (interrogative pronoun); the "E" will be chosen by the tagger since "K" cannot follow a noun ("N").
  - **R V** (article; verb): for example "l'appelle" (call him/her). The word "appelle" can only be a verb, but "l"' can be either an article or a personal pronoun. Thus, the rule will eliminate the article tag, giving preference to the pronoun.

4.2 Formulation of the tagging problem

An initial tag assignment is given on which a tagging scheme is applied through processing operators $P_1 P_2 ... P_n$. The goal is to obtain TA_n (a new tag assignment) with a maximal accuracy. That is, one wants to have $a(TA_n) = max$. Since there are many possible tagging schemes, one objective is to determine which one of them is the best. This will be the "optimal tagging scheme" which will be kept for tagging.

5 Implementation

We have developed all the software tools necessary in preprocessing and tagging the text, as well as additional utility programs. Most of the tools are implemented in PERL and shell script.

Several software tools have been developed in PERL, with a few shell scripts, which execute the different operators described above, as well as additional bookkeeping filters, utilities, and other programs. These tools are described in detail in Appendix A. The different tools are used to implement the processing operators mentioned in the previous section.

5.1 Text preprocessing

A raw corpus of text is the input to the preprocessor. Several filters need to be applied in order to normalize the text. The following steps are applied:

- **Sentence boundaries:** places where sentences begin and end are identified and replaced by appropriate SGML tags. Punctuation symbols are also assigned special tags.

- **Proper nouns:** the morphological dictionary contains common nouns and proper nouns, but the productivity of proper nouns is very high. Therefore, each word starting a sentence needs to be identified and recognized as either a common or a proper noun. These words undergo special treatment: each word starting a sentence will be given the PROPER noun tag; after morphological analysis, if the word inherits a new analysis, the latter one will prevail; if not, the word is identified as PROPER noun and is dynamically added to the PROPER_NAMES dictionary (see Section 5.2). If an initial uppercase word is found in the middle of a sentence, it will inherit immediately the PROPER noun tag. An additional difficulty due to the accents appears. In continental French, accented characters lose their accents if they become capitalized. This is valid in both sentence initial position and in the middle of the sentence. Therefore, many words in the text will be missing their accents. A phonology-based recovery technique is applied in order to attempt to recover these accents. Namely, an initial uppercase vowel will get an accent if it precedes a consonant in the following configuration:
  - if the word starts with the following pattern ECV, where E is the character "E", C is one of the consonants [b, bl, br, c, ch, cl, cr, d, dl, dr, f, fl, fr, g, gl, gr, h, j, j, l, m, n, p, ph, pl, pr, q, r, s, sl, sr, t, tl, tr, v, vl, vr, z], and V one of the vowels [a, e, i, o, u, y], the acute accent is recovered.
  - if the observed word is "A" or "Etre", the accent will be grave or circumflex respectively in order to produce "à" and "être".
- **Acronyms:** similarly to the case for proper nouns, the an initial guess that a certain word might be an acronym will be validated only if there are no other tags available from the morphology lookup.
- **Compound words:** compound words or non-compositional words in French are to be tagged as a separate entity and not as a sequence of two or three different words. These are recognized as such by looking in a dictionary of lexical compounds at this stage and considered as a single lexical unit. For example, locutions such as "a priori", "top secret", or "raz de marée" will be treated as unique lexical entries.
- **Personal pronouns:** if two words are connected by a dash "-", and the second word is a personal pronoun, the two words are considered individually. For example, the compound "dit-elle" (said she) is analyzed as two words "dit" and "elle".
- **Word splitting:** when all preprocessing has completed, the corpus is split into words and translated from 8-bit characters to 7-bit ascii characters. Accents are expressed by diacritic symbols that follow the unaccented letter. For example "être" is represented as "e^tre".

5.2 Morphological processing

We use finite-state transducers (FST) for the morphological analysis. Our FST is built on the model developed for Spanish morphology [14] and handles mainly inflectional morphology as well as some derivational affixes, such as "anti-" (anti) in "anti-iranien" (anti-iranien), and "arrière" (great) in "arrière-grand-père" (great-grandfather). The finite-state dictionary is originally built using the Robert Encyclopedic dictionary [7] and is increased through acquisition of proper nouns from unrestricted texts. The FST used in the morphological stage of the tagger can consist of up to 4 distinct sub-FST's:

1. main (non-proper-noun FST),
2. proper-noun FST, compiled from various sources,
3. proper-noun FST dynamically generated from the training corpus,

4. proper-noun FST generated heuristically from the current test corpus.

Nearly complete conjugations for French verbs are included in the main FST.

5.3 Tagset choice and hand tagging

We believed that a flexible tagset will be of benefit for the diverse applications that could make use of the tagger. Thus we have provided a facility to translate between our original (large) tagset and the tagset in use for a specific application. We perform the deterministic stage (see below) on the large tagset in order to be able to disambiguate as many words as possible, and allow for a tagset switch at any time after the last deterministic operator in the tagging scheme. It turns out that whereas deterministic operators work better on the large tagset, it is unclear whether the statistical learning performs better on the small tagset.

Manual tagging was done on 10,000 words and used for the training corpus (for learning), and on the test corpus (for evaluating). We have provided a tool which prompts the user with a list of all tags from the possible tags for a given word and lets the user either choose the correct tag, or specify some additional tags if necessary.

5.4 Application of deterministic rules

Linguistic knowledge was utilized in the tagger in terms of negative constraints. It is more feasible for the computational linguist to predict forbidden transitions between tags rather than anticipate all the possibilities of that transition in the given language. The constraints are read from left to right and disallow a particular bigram or trigram of tags.

Examples: [Article Verb] states that a verb cannot follow an article.

Negative constraints can be gathered using four methods:

1. the literature,
2. linguistic knowledge,
3. manual analysis of tagged corpora,
4. automated learning.

In our current work, we have used the first three methods only.

During each iteration of the deterministic stage, anchors are identified. An anchor is a word which in the current tag assignment has only one possible tag. If a word is left with one tag only after the application of a negative rule, this word will be consequently used as an anchor for the next iteration. If the neighboring words and the anchor itself follow some pattern which is disallowed as a negative constraint, action is then taken. We have determined empirically that three iterations are sufficient for disambiguation of the sentence. The user can change the number of iterations if this becomes necessary. In the future we might consider an alternative approach in the propagation of negative constraints.

It is interesting to note that the list of negative constraints could be expanded much more if we were to ignore that some negative constraints fail in only a limited number of cases. For example, the negative constraint [N] [N] (noun followed by another noun) would fail only in a few special situations (namely "dimanche soir" and similar temporal constructs) for French.

For proper nouns and acronyms, we have adopted a heuristic approach: if we encounter a word with initial uppercase, we assume that it is a possible proper noun and add a "proper noun" tag to its genotype. Similarly, if the word has all uppercase characters, the word is a possible acronym and is given the appropriate tag. Later, after applying the deterministic operator, it is possible that a given tag (other than "proper noun" and "acronym") is ruled out due to negative constraints. Then the "proper noun" or "acronym" tag will remain.

5.5 Statistical learning of genotype resolutions

At this stage we try to identify linguistic phenomena according to which a certain genotype has a predominant "gene" (tag). It turns out that most of the genotypes have predominant "genes". Thus it is possible to resolve some ambiguities using the genotype decision for the genotype of the word by looking up at table of the most likely tags for certain genotypes. Such a table can be compiled from the training corpus. A measure of confidence has been used to apply decisions under a certain threshold. Table 3 shows the decisions made upon the application of the threshold.

genotype	decision	freq. f/n	strength
NMP P	P	82/82	98.54
BD3S NMS RDF	RDF	172/173	98.44
BD3S RDM	RDM	195/199	96.70
DMS NMS NXP RIMS W	RIMS	107/109	96.30
P RP	P	768/793	96.16
NMS pMS	pMS	30/30	96.09
NXP W	W	90/92	95.63
NMP V2SPI V2SPS	NMP	25/25	95.33

Table 3: Best decisions that can be made according to unigram distributions

We use a *strength* score for each statistical rule based on the frequency, f, of the decision among n observations of the tag genotype. For instance, Table 2 gives $f = 195$ and $n = 199$ for the decision RDM from the tag genotype $[BD3S, RDM]$. The strength score assumes that f results from a binary distributiuon $B(p, n)$. This is the distribution which results when n independent trials are made, each having probability p of the decision (and probability $1-p$ of any other member of the tag genotype). We do not know p, but must make an estimate from the data. When $\hat{p}$ is estimated as the proportion f/n of the decision in the tag genotype, then the theory of the binomial distribution ([12] ,page 398) gives

$$sd(\hat{p}) = \sqrt{p(1-p)/n}$$

We estimate

$$\hat{p} = \frac{f + 0.5}{n + 1}$$

so that neither $\hat{p}$ nor $(1 - \hat{p})$ will be zero. This procedure is explained in [3] ,pages 34–36. We can estimate the uncertainty of $\hat{p}$ by:

$$\sqrt{\hat{p}(1-\hat{p})/n}$$

and we use the strength

$$strength = (\hat{p} - \sqrt{\frac{\hat{p} * (1-\hat{p})}{n}}) * 100$$

for the decision. This score represents our estimate of the probability less our estimate of the uncertainty. Notice in the above table that 25/25 has a lower strength than 30/30 which in turn has a lower strength that 82/82. The strength measure is designed to give lower values for the same f/n the smaller n is. Several examples of genotype decisions obtained through statistical means are shown in table 5.

5.6 Application of the genotype resolutions

We do not necessarily want to use all genotype decisions. One can observe that by varying the number of decisions made on a genotype basis, we can obtain significantly different results. Therefore we have established a parameter for the A stage which shows which decisions to use. A certain genotype decision will be applied only if its strength is above the threshold. We have made evaluations using the following values for the threshold: from 5 (practically all decisions) to 30, 45, 60, 75, 90, and 100 (no decisions at all). The results summarizing the effect of the thresholds are shown in next section. This stage can preserve some ambiguous words if not all possible genotypes were present in the training corpus.

5.7 Translation between the large set of tags and the small set of tags

Since we use an internal (large) tagset for most of the disambugation, we can apply at some point a tagset reduction operator, which would collapse the large tagset into a smaller set of tags. The smaller set of tags is either the one predefined in the system, or a tagset given by the user of the system.

6 Analysis and evaluation of the method

6.1 Training and test corpora

We have chosen the following as our corpora:

- Training: 10,000 words from the ECI (European Corpus initiative) corpus.
- Test: 1,000 words from randomly chosen sentences in the AFP (Agence France Presse) corpus.

These corpora have a significant number of typographical errors and misprints. Typos can cause problems for two reasons:

- **at the deterministic stage:** if they become anchors, they can trigger incorrect removals of neighboring tags.
- **at the statistical stage:** they can lead to incorrect values of some genotype decisions.

6.2 Cross-validation

In order to evaluate the statistical consistency of our results, we performed a validation consisting of the following: we split the test corpus into 11 slices of equal length. 10 of them were extracted from one corpus, and the 11th one was extracted from a different corpus (different source, different subject material). We performed a series of training experiments, each time using 10 of the 11 slices for training and the remaining 11th slice for testing. It was statistically impossible to distinguish the performance of the tagger in the special case (when training occurred on one 10-slice corpus, and testing on the remaining 11th slice) from the other 10 experiments. More precisely, the performance of the tagger in the special case ranked 4th among the 11 experiments.

6.3 Technical characteristics of the system

1. Time complexity: all filters run linearly with the size of the test corpus,
2. System requirements: all software included in the tagger toolkit is written in Perl version 4, as well as in Bourne and C Shell script languages. The tagger should work on most Unix platforms.

7 Results

We have analyzed 43 tagging schemes, ranging from the morphology stage only (M) to a complex series of procedures [morphology-deterministic-statistical (with a threshold of 30)-deterministic-tagset reduction], or $(M)DA_{30}DT$.

7.1 Optimal Tagging Scheme

We have determined empirically that, under the current model, the best tagging scheme is $(M)DA_5T$, i.e. [morphology-deterministic-statistical (with a threshold of 5)-tagset reduction] as shown in Table 4.

In the following subsections we identify the factors that influence the accuracy of the tagging scheme.

7.2 Analysis by sequence

Table 4 demonstrates that at the end of the morphological stage, 53% of the corpus has a single, unique, and correct tag, 1% of the words is incorrectly tagged and 47% is still ambiguous. The deterministic stage increases the percentage of correct tags by almost 7% while the statistical stage with the maximum coverage (i.e. 5) provides almost 90% of correct tags. Various tagging schemes have quite different performance as Table 4 shows.

tagging scheme	%correct	% incorrect	%ambiguous
(M)	53.5	1.0	45.7
(M)D	60.9	1.2	38.1
(M)DA_5	89.3	9.4	1.3
(M)DT	64.7	1.3	34.1
(M)T	57.3	1.1	41.7
(M)DA_5T	90.4	8.4	1.2
(M)DA_5DT	90.2	8.7	1.1
(M)$A_{90}DT$	74.7	1.4	23.9
(M)TA_5	90.0	8.9	1.1
(M)$DA_{90}DT$	78.4	1.8	19.9

Table 4: Results of the different tagging schemes

The best scheme is the one that applies sequentially Morphology (M) , Negative Constraints (with 3 iterations) (D), Statistical Decisions with maximal coverage (A_5) , and Tag Reduction (T).

7.3 Analysis by threshold

Table 5 reflects the differences in performance of the tagger when only the threshold of the statistical operator varies. A lower value of the threshold represents more (possibly incorrect) statistical decisions; a higher value – fewer (but more reliable) decisions.

tagging scheme	%correct	% incorrect	%ambiguous
$(M)DA_5T$	90.4	8.4	1.2
$(M)DA_{30}T$	89.8	8.3	1.9
$(M)DA_{45}T$	89.1	7.9	3.0
$(M)DA_{60}T$	83.4	3.9	12.7
$(M)DA_{75}T$	81.8	2.7	15.6
$(M)DA_{90}T$	76.8	1.6	21.7
$(M)DA_{100}T$	53.5	1.0	45.7

Table 5: Analysis of statistical decisions

7.4 Analysis by tagset

Table 6 presents the different tagging schemes with reduction to the small set of tags at different levels. Because of the large discrepancy in number between the large tagset (253) and the small one (67), we hypothesized that the there might be a significant difference at each time the tagset was reduced. The numbers in Table 6 do not verify this hypothesis, and in fact show that the difference in performance is small when using different versions of the tagset.

tagging scheme	%correct	%incorrect	%ambiguous
(M)	53.5	1.0	45.7
$(M)T$	57.3	1.1	41.7
$(M)A_5$	89.1	9.7	1.2
$(M)A_5T$	90.2	8.7	1.1
$(M)DA_{90}$	73.3	1.6	25.3
$(M)DA_{90}T$	76.8	1.6	21.7
$(M)A_5D$	88.7	10.3	1.0
$(M)A_5DT$	89.8	9.3	0.9

Table 6: Comparison between the two tagsets

8 Applications

There are several ways one can think of using a part-of-speech tagger:

- text-to-speech synthesis: several levels of the text-to-speech at the grapheme-to-phoneme level, knowing the part-of-speech of a word can determine its pronunciation; for example, in the French sentence presented in Table 7 the words "président", "est" and "couvent" have a different pronunciation when they are an inflected verb or a noun.

  Also, at the duration level, studies such as [2] and [13] have shown that duration of function words tends to be shorter than non function words; therefore, a part-of-speech tagger can help finding these function words.
- querying tagged corpora can be very useful for studying collocations or bilingual correspondences [8] . For example, in [8] , a tagger for English [11] is utilized to disambiguate English

Sentence:	le	président	s'	est	arrêté	pour	parler	au	couvent
p.o.s.:	pron	noun verb	pron	aux verb noun	p. part. noun	prep	verb noun	prep-art	noun verb
pronunciation:	/lə/	/prezidɑ̃/ /prezid(ə)/	/s/	/e/ /ɛst/	/arete/	/pur/	/parle/	/o/	/kuvɑ̃/ /kuv(ə)/
translation:	the	president			stopped		speaking	at the	convent

Table 7: French sentence with prononuciation varying with the part-of-speech

text in order to determine verbs and non verbs. As the study is focused on correspondences between French and English motion verbs, the tagger marks the English verbs so that the French corresponding sentence is selected as a candidate for the analysis of bilingual correspondence.

9 Conclusion

We described a part-of-speech tagger that correctly tags over 91% of unrestricted text with a very small amount of training data. When the correct answer is not certain, the tagger keeps the remaining ambiguities. The use of linguistic knowledge and statistical learning is an original contribution to the disambiguation problem. A flexible tagset allows adaptation of the tagger for various natural language applications. Several questions, such as tagging unknown words and typographical errors, need to be solved. We are in the process of collecting more training data to improve the system performance as well as trying the tagger on other languages.

10 Acknowledgments

We would like to thank Ido Dagan and Diane Lambert for the comments, suggestions, and support that they provided throughout the work.

A User's Manual: description of the MT toolset

We have developed a series of tools which can be reused in other similar problem set-ups. Each tool is a stand-alone utility, and pipelines of such tools can be designed to perform various tasks.

There are 4 directories where the tagger and the corpora reside: ${TAGGERDIR}, ${TRAININGDIR}, ${TESTDIR}, ${TEMPDIR}.

In order to tag a corpus, the user needs to perform the following steps:

- know where the system files are located.
- create a directory and put the corpus file in it. The extension of the file should be .cor
- copy the system makefile into the directory where the corpus is located and modify it so that the values of the directories are set properly.
- modify the environmental variable "CORPUS" to designate the name of the corpus file. E.g. if the corpus file is called MYCORPUS.cor, then the user has to set CORPUS="MYCORPUS"
- type "make MYCORPUS.DA5T" for the best tagging sequence. Any other tagging sequence can be obtained by replacing "DA5T" in the previous command by the corresponding tagging sequence acronym.

A.1 System library files

- NCONS3: list of negative constraints
- NOSP: list of compound words
- TAGS: mapping between the large and small tagsets
- arclistd: finite-state transducer for morphological analysis
- MAINPROPERS.arclist: finite-state transducer that contains many proper nouns

A.2 Morphological analyzer

- dictionary: finite-state transducer driver

A.3 Makefile

- makefile: script that is used for tagging

A.4 Filters which are part of the tagger itself

- mtapply: puts together the tags resulting from applying 1-gram and bigram statistical decisions
- mtback: translates the output of mtiter into the normal tag assignment format.
  Example: ":P:NP:#:NMS:NFS:" becomes "P NP" followed on the next line by "NMS NFS"
- mtcompound: this filter checks for compounds in the input and outputs them as a single token.
  Example: if "de", "façon", and "que" appear in the input, the output will contain "de_façon_que"
- mtconcise: this filter translates the verbose morphological features and parts of speech from the FST into concise tags from the tagset.
  Example: "noun masc. plur." becomes "NMP"
- mthsuniq: removes duplicate tags
- mtiter: applies the negative constraints on a tag assignment
- mtlearn: statistically computes the best statistical decisions from the training corpus
- mtnop: removes the "proper noun" and "acronym" tags if others tags are present for the same word
- mtnosgml: removes SGML tags from the input corpus.p
- mtpn: handles pronouns in constructions such as "dit-elle"
- mtprintl: print all tags for a given word on the same line
- mtrestore: recovers tags that have been ruled out at some stage
- mtsplit: splits the corpus into a list of the words in in
- mtstat: applies the statistical decisions
- mttest2: computes the accuracy of the tagging when given the correct tagging
- mttrans: translates the large tagset into the small tagset

A.5 Other tools

The following tools are used mostly for debugging.

- mtasc: changes 7-bit French text to 8-bit text.
- mtbatch: batch-mode utility.
- mtcount: counts the ambiguities in a given tag assignment.
- mteval: batch-mode utility.
- mtex: batch-mode utility.
- mthuniq: same as mthuniq, but assumes that the tags on each line of the input are sorted.
- mtlearn2: same as mtlearn, but it also uses genotype bigrams.
- mtlc: converts the input into lowercase.
- mtnop.s: same as mtnop, but works on the small tagset.
- mtrun: batch-mode utility.
- mtselect: utility for manual tagging.
- mtshow-allstages: visualization utility
- mtshow-detstage: visualization utility
- mtshow-disambig: visualization utility
- mtshow-wrong: visualization utility
- mttop: shows the most frequent words in a corpus.

B Choosing a Tagset

The following list shows the tagsets that are used in the system. The first column indicates the restricted set of tags, and the second column indicates the extended set of tags. Notice that the user can specify any subset of tags being contained in the large set. In order to specify a different set, map the new tag to the large one, and write the change in the first column.

AT&T

Document Cover Sheet for Technical Memorandum

Title: Part-of-Speech Tagger for French: a User's Manual

Authors	**Electronic Address**	**Location**	**Ext.**	**Company** (if other than AT&T–BL)
Dragomir R. Radev	s_radev@research.att.com	MH 2D-468	4078	
Evelyne Tzoukermann	evelyne@research.att.com	MH 2D-448	2924	
William A. Gale	gale@research.att.com	MH 2C-278	2520	

Document No.	**Filing Case No.**	**Work Project No.**
11222-950726-03TM	60011	311402-2228
11215-950727-08TM	20878	311401-1503

Keywords:
Text-to-Speech Synthesis; French Text Analysis; Part-of-Speech Tagging; Computational Morphology

MERCURY Announcement Bulletin Sections
CMM-Communications CMP-Computing CFS-Life Sciences

Abstract

The purpose of this work is to produce a part-of-speech tagger for French using morphological analysis provided by a finite-state transducer. The tagger also utilizes a combination of statistical learning and linguistic knowledge and is built in a pipelined architecture. All modules, except for preprocessing and morphological analysis, can be ordered in various ways. Part of speech tagging consists of applying several disambiguation modules on a list of ambiguous words until a single tag remains for each word. We propose and evaluate a sequencing strategy for the various modules and point out the best sequencing available. Several experiments were performed to figure out the best order of the different modules. Results showed that optimal results are obtained when morphological analysis is applied first, followed, in that order, by the application of linguistic constraints, the statistical stage, and, finally, the mapping of the large tagset to a smaller one. The system works on unrestricted text.

Pages of Text 1 Other Pages 15 Total 16
No. Figs. 0 No. Tables 7 No. Refs. 0

Mailing Label

Complete Copy

DH 1122
MTS 11222
Kenneth W. Church
Cathy Cohen
Eileen Fitzpatrick
Julia Hirschberg
Donald Hindle
James Hieronymus
Mark Jones
Diane Lambert
David Lewis
Fernando Pereira
Lawrence R. Rabiner
Thomas Restaino
David Yarowsky

Cover Sheet Only

Arno Penzias
1122 MTS

Future AT&T Distribution by ITDS Release to any AT&T employee (excluding contract employees)

Author Signatures

Dragomir R. Radev | Evelyne Tzoukermann | William A. Gale

Organizational Approval: (Department Head)

Steve E. Levinson

For Use by Recipient of Cover Sheet:

Computing network users may order copies via the *library -k* command;
for information, type "man library" after logon.
Otherwise:

1 Enter PAN if AT&T–BL (or SS# if non-AT&T–BL). _____
2 Fold this sheet in half with this side out.
3 Check the address of your local Internal Technical Document Service if listed; otherwise, use HO 4F-112. Use no envelope.
4 Indicate whether microfiche or paper copy is desired.

Internal Technical Document Service

( ) ALC 1B-102A ( ) IH 7M-103
( ) CB 3O-2011 ( ) MV 3L-19
( ) HO 4F-112 ( ) WH 3E-204

Please send a complete ☐ microfiche ☐ paper copy of this document to the address shown on the other side.

Contents

SHORT_SET	LARGE_SET	MEANING OF THE TAG
v1p	&1PPI	auxiliary 1st person plural present indicative
v1p	&1PPM	auxiliary 1st person plural present imperative
v1p	&1PPC	auxiliary 1st person plural present conditional
v1p	&1PPS	auxiliary 1st person plural present subjunctive
v1p	&1PFI	auxiliary 1st person plural future indicative
v1p	&1PII	auxiliary 1st person plural imperfect indicative
v1p	&1PSI	auxiliary 1st person plural simple-past indicative
v1p	&1PIS	auxiliary 1st person plural imperfect subjunctive
v2p	&2PPI	auxiliary 2nd person plural present indicative
v2p	&2PPM	auxiliary 2nd person plural present imperative
v2p	&2PPC	auxiliary 2nd person plural present conditional
v2p	&2PPS	auxiliary 2nd person plural present subjunctive
v2p	&2PFI	auxiliary 2nd person plural future indicative
v2p	&2PII	auxiliary 2nd person plural imperfect indicative
v2p	&2PSI	auxiliary 2nd person plural simple-past indicative
v2p	&2PIS	auxiliary 2nd person plural imperfect subjunctive
v3p	&3PPI	auxiliary 3st person plural present indicative
v3p	&3PPC	auxiliary 3rd person plural present conditional
v3p	&3PPS	auxiliary 3rd person plural present subjunctive
v3p	&3PFI	auxiliary 3rd person plural future indicative
v3p	&3PII	auxiliary 3rd person plural imperfect indicative
v3p	&3PSI	auxiliary 3rd person plural simple-past indicative
v3p	&3PIS	auxiliary 3rd person plural imperfect subjunctive
v1s	&1SPI	auxiliary 1st person singular present indicative
v1s	&1SPM	auxiliary 1st person singular present imperative
v1s	&1SPC	auxiliary 1st person singular present conditional
v1s	&1SPS	auxiliary 1st person singular present subjunctive
v1s	&1SFI	auxiliary 1st person singular future indicative
v1s	&1SII	auxiliary 1st person singular imperfect indicative
v1s	&1SSI	auxiliary 1st person singular simple-past indicative
v1s	&1SIS	auxiliary 1st person singular imperfect subjunctive
v2s	&2SPI	auxiliary 2nd person singular present indicative
v2s	&2SPM	auxiliary 2nd person singular present imperative
v2s	&2SPC	auxiliary 2nd person singular present conditional
v2s	&2SPS	auxiliary 2nd person singular present subjunctive
v2s	&2SFI	auxiliary 2nd person singular future indicative
v2s	&2SII	auxiliary 2nd person singular imperfect indicative
v2s	&2SSI	auxiliary 2nd person singular simple-past indicative
v2s	&2SIS	auxiliary 2nd person singular imperfect subjunctive
v3s	&3SPI	auxiliary 3st person singular present indicative
v3s	&3SPC	auxiliary 3rd person singular present conditional
v3s	&3SPS	auxiliary 3rd person singular present subjunctive
v3s	&3SFI	auxiliary 3rd person singular future indicative
v3s	&3SII	auxiliary 3rd person singular imperfect indicative
v3s	&3SSI	auxiliary 3rd person singular simple-past indicative
v3s	&3SIS	auxiliary 3rd person singular imperfect subjunctive
v	&N	auxiliary infinitive
qp	&QP	auxiliary present participle
qp	&QPMS	auxiliary present participle masculine singular
qs	&QS	auxiliary past participle
qsfp	&QSFP	auxiliary past participle feminine plural
qsfs	&QSFS	auxiliary past participle feminine singular
qsmp	&QSMP	auxiliary past participle masculine plural
qsms	&QSMS	auxiliary past participle masculine singular
a	A	adverb

SHORT_SET	LARGE_SET	MEANING OF THE TAG
b	BI	indefinite personal pronoun
bfp	BD3FP	personal pronoun direct feminine 3rd person plural
bfs	BD3FS	personal pronoun direct feminine 3rd person singular
bmp	BD3MP	personal pronoun direct masculine 3rd person plural
bms	BD3MS	personal pronoun direct masculine 3rd person singular
b	BD1P	personal pronoun direct 1st person plural
b	BD1S	personal pronoun direct 1st person singular
b	BD2P	personal pronoun direct 2nd person plural
b	BD2S	personal pronoun direct 2nd person singular
b	BD3P	personal pronoun direct 3rd person plural
b	BD3S	personal pronoun direct 3rd person singular
bfp	BI3FP	personal pronoun indirect feminine 3rd person plural
bfs	BI3FS	personal pronoun indirect feminine 3rd person singular
bmp	BI3MP	personal pronoun indirect masculine 3rd person plural
bms	BI3MS	personal pronoun indirect masculine 3rd person singular
b	BI1P	personal pronoun indirect 1st person plural
b	BI1S	personal pronoun indirect 1st person singular
b	BI2P	personal pronoun indirect 2nd person plural
b	BI2S	personal pronoun indirect 2nd person singular
b	BI3P	personal pronoun indirect 2nd person plural
b	BI3S	personal pronoun indirect 2nd person singular
bfp	BJ3FP	personal pronoun disjoint feminine 3rd person plural
bfs	BJ3FS	personal pronoun disjoint feminine 3rd person singular
bmp	BJ3MP	personal pronoun disjoint masculine 3rd person plural
bms	BJ3MS	personal pronoun disjoint masculine 3rd person singular
b	BJ1P	personal pronoun disjoint 1st person plural
b	BJ1S	personal pronoun disjoint 1st person singular
b	BJ2P	personal pronoun disjoint 2nd person plural
b	BJ2S	personal pronoun disjoint 2nd person singular
b	BJ3P	personal pronoun disjoint 2nd person plural
b	BJ3S	personal pronoun disjoint 2nd person singular
bfp	BR3FP	personal pronoun reflechi feminine 3rd person plural
bfs	BR3FS	personal pronoun reflechi feminine 3rd person singular
bmp	BR3MP	personal pronoun reflechi masculine 3rd person plural
bms	BR3MS	personal pronoun reflechi masculine 3rd person singular
b	BR1P	personal pronoun reflechi 1st person plural
b	BR1S	personal pronoun reflechi 1st person singular
b	BR2P	personal pronoun reflechi 2nd person plural
b	BR2S	personal pronoun reflechi 2nd person singular
b	BR3P	personal pronoun reflechi 3rd person plural
b	BR3S	personal pronoun reflechi 3rd person singular
bfp	BS3FP	personal pronoun subject feminine 3rd person plural
bfs	BS3FS	personal pronoun subject feminine 3rd person singular
bmp	BS3MP	personal pronoun subject masculine 3rd person plural
bms	BS3MS	personal pronoun subject masculine 3rd person singular
b	BS1P	personal pronoun subject 1st person plural
b	BS1S	personal pronoun subject 1st person singular
b	BS2P	personal pronoun subject 2nd person plural
b	BS2S	personal pronoun subject 2nd person singular
cc	CC	coordinating conjunction
cs	CS	subordinating conjunction
b	D	indefinite pronoun
b	DFS	indefinite pronoun feminine singular
b	DFP	indefinite pronoun feminine plural
b	DMP	indefinite pronoun masculine singular

SHORT_SET	LARGE_SET	MEANING OF THE TAG
b	DMS	indefinite pronoun masculine singular
b	DP	indefinite pronoun plural
b	E	relative pronoun
bf	EF	relative pronoun feminine
bfp	EFP	relative pronoun feminine plural
bm	EM	relative pronoun masculine
bmp	EMP	relative pronoun masculine plural
bfs	GFS	possessive pronoun feminine singular
bfp	GFP	possessive pronoun feminine plural
bmp	GMP	possessive pronoun masculine plural
bms	GMS	possessive pronoun masculine singular
bp	GP	possessive pronoun plural
bs	GS	possessive pronoun singular
i	I	interjection
jfp	JFP	feminine plural adjective
jfs	JFS	feminine singular adjective
jmp	JMP	masculine plural adjective
jms	JMS	masculine singular adjective
jm	JMX	masculine adjective invariable in number
jp	JXP	plural adjective invariable in gender
js	JXS	singular adjective invariable in gender
j	JXX	invariable adjective
jp	JP	plural adjective
js	JS	singular adjective
k	K	interrogative pronoun
kf	KF	interrogative pronoun feminine
kfp	KFP	interrogative pronoun feminine plural
km	KM	interrogative pronoun masculine
kmp	KMP	interrogative pronoun masculine plural
b	L	pronoun
b	L3S	pronoun 3rd person singular
b	LFP	pronoun feminine plural
b	LFS	pronoun feminine singular
b	LMP	pronoun masculine plural
b	LMS	pronoun masculine singular
b	LXP	pronoun plural invariable in gender
b	LXS	pronoun singular invariable in gender
nf	NF	feminine noun
nfp	NFP	feminine plural noun
nfs	NFS	feminine singular noun
nf	NFX	feminine noun invariable in number
nm	NM	masculine noun
nmp	NMP	masculine plural noun
nms	NMS	masculine singular noun
nm	NMX	masculine noun invariable in number
n	NXS	singular noun invariable in gender
n	NXP	plural noun invariable in gender
n	NXX	invariable noun
o	O	onomat.
p	P	preposition
qp	QP	present participle
qp	QPMS	present participle masculine singular
qs	QS	past participle
qsfp	QSFP	past participle feminine plural
qsfs	QSFS	past participle feminine singular

SHORT_SET	LARGE_SET	MEANING OF THE TAG
qsmp	QSMP	past participle masculine plural
qsms	QSMS	past participle masculine singular
r	R	article
r	RD	definite article
rf	RDF	definite feminine article
rm	RDM	definite masculine article
rm	RDMP	definite masculine plural article
rm	RDMS	definite masculine singular article
r	RDP	definite partitive article
r	RI	indefinite article
r	RIFS	indefinite feminine singular article
r	RIFP	indefinite feminine plural article
r	RIMP	indefinite masculine plural article
r	RIMS	indefinite masculine singular article
r	RP	partitive article
i	S	particle
a	T	nominal
u	U	proper noun
v1p	V1PPI	verb 1st person plural present indicative
v1p	V1PPM	verb 1st person plural present imperative
v1p	V1PPC	verb 1st person plural present conditional
v1p	V1PPS	verb 1st person plural present subjunctive
v1p	V1PFI	verb 1st person plural future indicative
v1p	V1PII	verb 1st person plural imperfect indicative
v1p	V1PSI	verb 1st person plural simple-past indicative
v1p	V1PIS	verb 1st person plural imperfect subjunctive
v2p	V2PPI	verb 2nd person plural present indicative
v2p	V2PPC	verb 2nd person plural present conditional
v2p	V2PPS	verb 2nd person plural present subjunctive
v2p	V2PFI	verb 2nd person plural future indicative
v2p	V2PII	verb 2nd person plural imperfect indicative
v2p	V2PSI	verb 2nd person plural simple-past indicative
v2p	V2PIS	verb 2nd person plural imperfect subjunctive
v3p	V3PPI	verb 3rd person plural present indicative
v3p	V3PPC	verb 3rd person plural present conditional
v3p	V3PPS	verb 3rd person plural present subjunctive
v3p	V3PFI	verb 3rd person plural future indicative
v3p	V3PII	verb 3rd person plural imperfect indicative
v3p	V3PSI	verb 3rd person plural simple-past indicative
v3p	V3PIS	verb 3rd person plural imperfect subjunctive
v1s	V1SPI	verb 1st person singular present indicative
v1s	V1SPM	verb 1st person singular present imperative
v1s	V1SPC	verb 1st person singular present conditional
v1s	V1SPS	verb 1st person singular present subjunctive
v1s	V1SFI	verb 1st person singular future indicative
v1s	V1SII	verb 1st person singular imperfect indicative
v1s	V1SSI	verb 1st person singular simple-past indicative
v1s	V1SIS	verb 1st person singular imperfect subjunctive
v2s	V2SPI	verb 2nd person singular present indicative
v2s	V2SPM	verb 2nd person singular present imperative
v2s	V2SPC	verb 2nd person singular present conditional
v2s	V2SPS	verb 2nd person singular present subjunctive
v2s	V2SFI	verb 2nd person singular future indicative
v2s	V2SII	verb 2nd person singular imperfect indicative
v2s	V2SSI	verb 2nd person singular simple-past indicative

SHORT_SET	LARGE_SET	MEANING OF THE TAG
v2s	V2SIS	verb 2nd person singular imperfect subjunctive
v3s	V3SPI	verb 3st person singular present indicative
v3s	V3SPC	verb 3rd person singular present conditional
v3s	V3SPS	verb 3rd person singular present subjunctive
v3s	V3SFI	verb 3rd person singular future indicative
v3s	V3SII	verb 3rd person singular imperfect indicative
v3s	V3SSI	verb 3rd person singular simple-past indicative
v3s	V3SIS	verb 3rd person singular imperfect subjunctive
v	i	verb infinitive
z	W	numeral
b	Y	demonstrative pronoun
bfp	YFP	demonstrative pronoun feminine plural
bfs	YFS	demonstrative pronoun feminine singular
bmp	YMP	demonstrative pronoun masculine plural
bms	YMS	demonstrative pronoun masculine singular
jfpd	dFP	demonstrative adjective feminine plural
jfsd	dFS	demonstrative adjective feminine singular
jmpd	dMP	demonstrative adjective masculine plural
jmsd	dMS	demonstrative adjective masculine singular
jmsd	dP	demonstrative adjective plural
jmsd	dS	demonstrative adjective singular
jfpp	pFP	possessive adjective feminine plural
jfsp	pFS	possessive adjective feminine singular
jmpp	pMP	possessive adjective masculine plural
jmsp	pMS	possessive adjective masculine singular
jp	pP	possessive adjective plural
js	pS	possessive adjective singular
x	.	punctuation
h	r	acronym
^	^	sentence beginning
$	$	sentence end
****	*****	NIL
???	???	ERROR

References

[1] Lalit R. Bahl and Robert L. Mercer. Part-of-speech assignement by a statistical decision algorithm. *IEEE International Symposium on Information Theory*, pages 88–89, 1976.

[2] K. Bartkova and C. Sorin. A model of segmental duration for speech synthesis in French. *Speech Communication*, 6:245–260, 1987.

[3] G.E.P. Box and G.C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison-Wesley, Reading. Mass., 1973.

[4] Eric Brill. A simple rule-based part of speech tagger. In *Third Conference on Applied Computational Linguistics*, Trento, Italy, 1992.

[5] Kenneth W. Church. A stochastic parts program noun phrase parser for unrestricted text. In *IEEE Proceedings of the ICASSP*, pages 695–698, Glasgow, 1989.

[6] Steve DeRose. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39, 1988.

[7] Alain Duval et al. *Robert Encyclopedic Dictionary (CD-ROM)*. Hachette, Paris, 1992.

[8] Judith Klavans and Evelyne Tzoukermann. Dictionaries and corpora: Combining corpus and machine-readable dictionary data for building bilingual lexicons. *Computational Linguistics*, , to appear. under review.

[9] S. Klein and R. F. Simmons. A grammatical approach to grammatical tagging coding of english words. *JACM*, 10:334–347, 1963.

[10] Geoffrey Leech, Roger Garside, and Erik Atewll. Automatic grammatical tagging of the lob corpus. *ICAME News*, 7:13–33, 1983.

[11] Bernard Merialdo. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–172, 1994.

[12] D.S. Moore and G.P. McCabe. *Introduction to the Practice of Statistics*. W. H. Freeman, New York, 1989.

[13] O. Soumoy, Tzoukermann E., and J. P. H. van Santen. Duration in french text-to-speech synthesis. In *11222-941202-18-TM*, Murray Hill, N.J., USA, 1994. Technical Memorandum, AT&T Bell Laboratories.

[14] Evelyne Tzoukermann and Mark Y. Liberman. A finite-state morphological processor for spanish. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki, Finland, 1990. International Conference on Computational Linguistics.

[15] Atro Voutilainen. Nptool, a detector of english noun phrases. Columbus, Ohio, 1993. Proceedings of the Workshop on very large corpora.